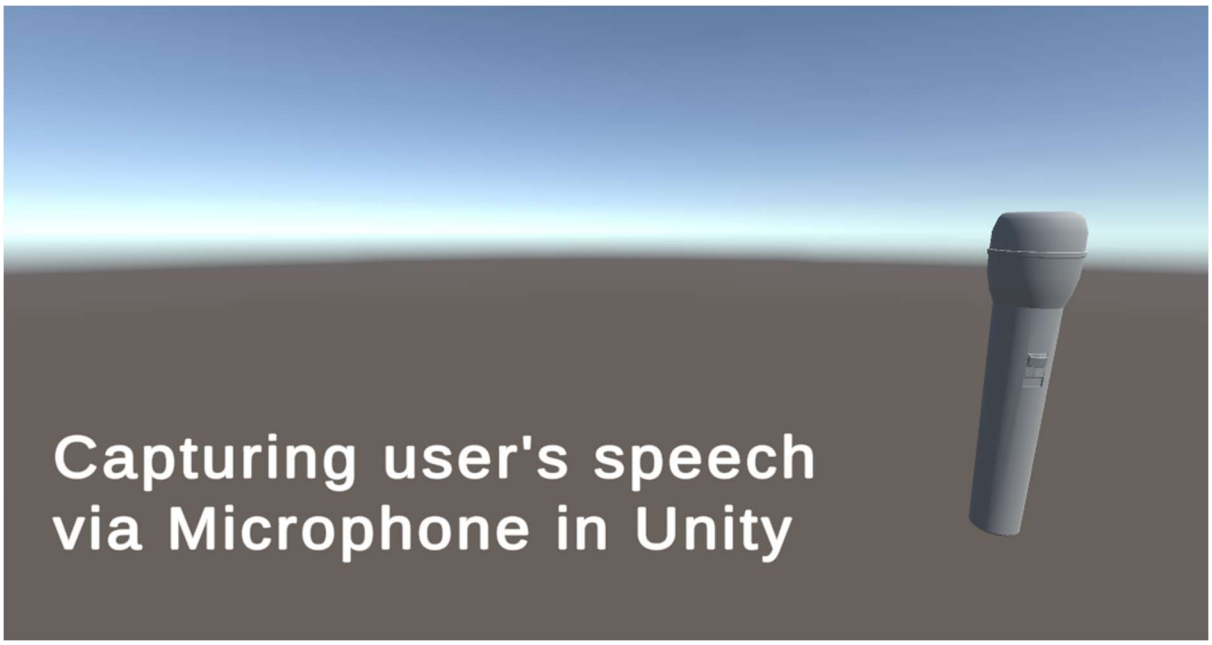
과제 #1 - Capturing User's Speech (Utterance) via Microphone in Unity

전자공학과 120220117 박준호

1. 목적

✓ 사용자의 음성을 마이크를 통해 입력하고, 이 음성을 스피커를 통해 real-time으로 출력하는 환경을 unity를 통해 구현



2. 코드 분석

✓ 변수 선언

■ 마이크 실행 및 음성 출력에 필요한 변수들 미리 선언

```
private AudioSource source;
private AudioClip mic;
private int lastSample = 0;
private float[] samples = null;
private List<float> readSamples = null;
private int channels = 0;
private int readUpdateId = 0;
private int previousReadUpdateId = -1;
public float READ_FLUSH_TIME = 0.5f;
private float readFlushTimer = 0.0f;
```

✓ void Start()

■ 스크립트가 시작할 때 1회 실행되는 함수

```
void Start()
{
readSamples = new List<float>();
source = new GameObject("Mic").AddComponent<AudioSource>();
mic = Microphone.Start("Microphone (High Definition Audio Device)",
true, 100, 44100
channels = mic.channels
}
```

- readSamples를 List 변수로 지정하고, 음성을 입력으로 받는 Mic라는 game object를 정의한다. 그리고 마이크의 디바이스 이름, loop 여부, 초당 받는 음성의 length, 주파수를 입력하여 mic라는 변수를 정의한다. 마지막으로 mic의 채널 수를 갸져와 channels라는 변수로 정의한다.

✓ void Update()

■ 스크립트가 실행되는 동안 매 프레임마다 실행시키는 함수

```
void Update()
{
        ReadMic();
        PlayMic();
}
```

- ReadMic를 통해 음성을 마이크를 통해 입력받고, PlayMic를 통해 입력받은 음성을 스피커를 통해 출력한다.

```
private void ReadMic()
{
        int pos = Microphone.GetPosition(null);
        int diff = pos - lastSample;
        if (diff > 0)
        {
                samples = new float[diff * channels];
                mic.GetData(samples, lastSample);
                readSamples.AddRange(samples);
        }
        lastSample = pos;
}
```

- Microphone의 GetPosition 메서드를 통해 녹음하는 샘플의 포지션을 얻어 pos에 넣는다. 이 값과 바로 이전 프레임에서의 pos와의 차이를 계산하여 diff로 정의하고, 만약 diff가 0보다 크면 현재 samples와 지난 lastSample을 readSamples에 저장한다.

```
private void PlayMic()
{
        readFlushTimer += Time.deltaTime;
        if (readFlushTimer > READ_FLUSH_TIME)
        {
```

```
            if (readUpdateId != previousReadUpdateId && readSamples !=
null && readSamples.Count > 0)
            {
                previousReadUpdateId = readUpdateId;
                source.clip = AudioClip.Create("Real_time",
readSamples.Count, channels, 44100, false);
                source.spatialBlend = 0;
                source.clip.SetData(readSamples.ToArray(), 0);
                if (!source.isPlaying){
                    source.Play();
                }
                readSamples.Clear();
                readUpdateId++;
            }
            readFlushTimer = 0.0f;
        }
}
```

- 한 프레임이 처리되는 시간(Time.deltaTime)을 readFlushTimer에 입력한 뒤, 기존에 정의한 값(READ_FLUSH_TIME)보다 크고, readUpdateId가 이전 PlayMic 함수에서의 readUpdateID(previousReadUpdateId)와 다르며, readSamples의 값이 비어있지 않고, 그 수가 0보다 크다면, 입력받은 음성 샘플을 출력하는 코드가 실행된다.

  출력 과정은, 우선 AudioClip의 Create라는 메서드를 통해 샘플의 수, 채널, 주파수를 입력받고, 이를 SetData 메서드로 스피커로 출력할 수 있게 정렬한 다음, Play 메서드로 음성을 출력하게 된다. 그 후 readSamples에 있는 샘플들을 Clear를 통해 비우고, readUpdateId를 1을 올려 업데이트하며, readFlushTimer 또한 0으로 초기화한다.

3. 기타

✓ Canvas

■ 첫 페이지의 게임 캡쳐 화면과 같이 텍스트를 Canvas에 추가하여 시각화

✓ MicGet

■ Unity Asset Store에서 마이크를 형상화한 asset을 구매 후 object로 추가하여 시각화